84K00220
June 16, 1997

Draft

Checkout and Launch Control System (CLCS) Concept of Operations

Version 0.0

Prepared By:

CLCS User Liaison Team

Kennedy Space Center, FL 32899

CONCEPT OF OPERATIONS
Table of Contents

1. Introduction 1
1.1. Scope 1
1.2. Operational Overview 1
1.3. Document Overview 3
1.4. Applicable Documents 3
2. Operations Concepts 3
2.1. Physical Characteristics 3
2.1.1. Launch Control Center (LCC) Set 3
2.1.2. Local Operation Definition 10
2.1.3. Specialized Processing Sites 11
2.1.4. Software Sustaining Environment 11
2.2. Roles and Responsibilities 12
2.2.1. Users 12
2.2.2. Operators 12
2.2.3. Sustainers (Developers) 12
3. CLCS Operations 12
3.1. Support Operations 12
3.1.1. Test Definition 13
3.1.2. Set Configuration and Start Up 13
3.1.3. Operational Readiness Test Task 14
3.1.4. Network Manager Function 14
3.1.5. Data Retrieval and Distribution 14
3.1.6. Simulation 15
3.1.7. Operational Maintenance 15
3.2. Test Operations 16
3.2.1. Test Control 16
3.2.2. Command Authority Management 16
3.2.3. Application Start and System Status 17
3.2.4. Command/Monitor Application 17
3.2.5. Consolidation Efforts 17
3.2.6. Data Recording 17
3.2.7. Data Access 18
3.2.8. Operations Redundancy Definition 18
3.2.9. System Integrity Task 18
3.2.10. Subsystem Integrity Task 19
3.2.11. Checkpoint Task 19
4. System Processes 19
4.1. Hardware 19
4.1.1. Hardware Maintenance 19
4.2. System Software 20
4.2.1. System Build & Load 20
4.2.2. Data Bank-System Build 21
4.3. Application Software 21
4.3.1. Simulation Software 22
4.3.2. Command, Control and Monitor (CCM) Application Software 22
4.3.3. Test Build and Load 23
5. User Environment 24
5.1. Safing 24
5.1.1. LDB Safing 24
5.1.2. Hardwire Safing 25
5.1.3. Programmable Function Panel (PFP) 25
5.1.4. Reactive Control Logic 25
5.2. Utility Tools 25

CONCEPT OF OPERATIONS
Table of Contents

5.2.1. End Item Manager Function 25
5.2.2. Data Fusion Function 26
5.2.3. Data Path Health Function 26
5.2.4. FD Constraint Check Function 26
5.2.5. Command Entry 26
5.3. Input Devices 27
5.3.1. Command/Business Network Separation 27
5.3.2. Pointing Devices 27
5.4. Output Devices 27
5.5. Display Characteristics/Capabilities 27
5.5.1. Message Window 28
5.5.2. Overlaying of Windows 28
5.6. Business Systems 28
5.6.1. World Wide Web/Internet Access 29
5.7. Advisory Systems 29
5.8. Operational Intercommunication System 30
5.9. Operational Television Displays 30
5.9.1. View Sequencing 30
5.9.2. Camera Control 30
5.9.3. Timing 30
5.9.4. Special Functions for Recorded Views 30
6. Operations Scenarios 31
6.1. Integrated Vehicle/Launch Environment 31
6.2. Non Integrated/Multi-flow Environment 32
6.3. Simulation Environment 33
7. Acronym List 34

1. Introduction

1.1. Scope

This document contains the concept of operations for the Checkout and Launch Control System (CLCS) and provides the basis from which the user communities system level requirements are derived for the Launch Control Center (LCC), specialized processing sites, and all activities required to define, prepare and run test operations. This document is specific to the replacement of LPS with CLCS, it does not address operational interfaces for future projects and new initiatives. Although, it is intended that CLCS will be developed and deployed in a flexible, adaptable ,and portable manner so as to accommodate future initiatives. Documents detailing CLCS operations for these upgrade projects will be produced as required.

1.2. Operational Overview

Shuttle processing at KSC can be categorized in two ways: 1) Flight element and Ground Support Equipment (GSE) processing; and 2) Integrated vehicle processing, including launch. These categories of processing have unique requirements which affect how we use CLCS. In addition, in order to continue processing successfully, CLCS requires ongoing support, maintenance, and sustaining efforts. Processing will be supported from three OCR's, each configured differently to support integrated and non-integrated flight hardware and ground equipment testing. Two of the control rooms could support both categories of testing. The third control room will be a smaller set which supports only non-integrated testing. Additionally, engineers will be able to control tests at selected sites locally with a plug-in portable workstation and retrieve SDS (Shuttle Data Stream) data for analysis from their office.

Day to day processing involves use of multiple control rooms, Cargo Integration Test Environment (CITE), Space Station Processing Facility (SSPF), Hypergolic Maintenance Facility (HMF), and Complex Control System (CCS) resources. With the wide range of test, flight element, and ground configurations needed in Shuttle processing, CLCS must be made both flexible and scaleable. As an example, a control room could be divided into two zones processing a non-integrated vehicle, the associated GSE, and an MLP (Mobile Launcher Platform) on one side of the room while the other side may only need to process an ET (External Tank) housed in VAB (Vehicle Assembly Building) high bay 2 checkout cell and the GSE at the non-launch pad. Similar testing may be occurring in all three control rooms on the same day with a mixture of flight elements, GSE and facilities.

Integrated vehicle testing begins when the orbiter is mated to its external tank and boosters in the VAB. The integrated vehicle, associated GSE and the launch pad systems supporting the vehicle will be controlled out of a single OCR. It is intended that testing on a single vehicle, GSE, and associated facilities are the only activities in an OCR once the vehicle becomes integrated. During the checkout and launch of the integrated vehicle on the launch pad, data is made available to other centers for review and analysis. This includes the HOSC (Huntsville Operations Support Center) at Marshall Space Flight Center (MSFC), the Mission Evaluation Room (MER) at Johnson Space Center (JSC), and other centers required for test support.

It is intended that a minimum level of support, maintenance, and sustaining activities will be necessary to keep CLCS operational. CLCS support activities include management and control of CLCS system resources and preparation and monitoring of Test Configuration Identifiers (TCID) in support of testing. CLCS maintenance activities include failure analysis, fault isolation, recovery, and preventative maintenance. Sustaining activities include configurations and updates for hardware, system software, application software, and data bank/system builds.

In addition to the Launch Control Center (LCC), off-line processing of different elements occurs daily at the HMF for the Orbiter Maneuvering System checkout and CITE/SSPF for payload checkout. The

Complex Control Center (CCC) monitors the facility functions like air conditioning, electrical power, Firex, and pneumatics around the clock. DFRC will also have a set for post landing processing when the orbiter lands there.

Figure 1-1 illustrates the wide distribution of CLCS data required across the center and the agency.

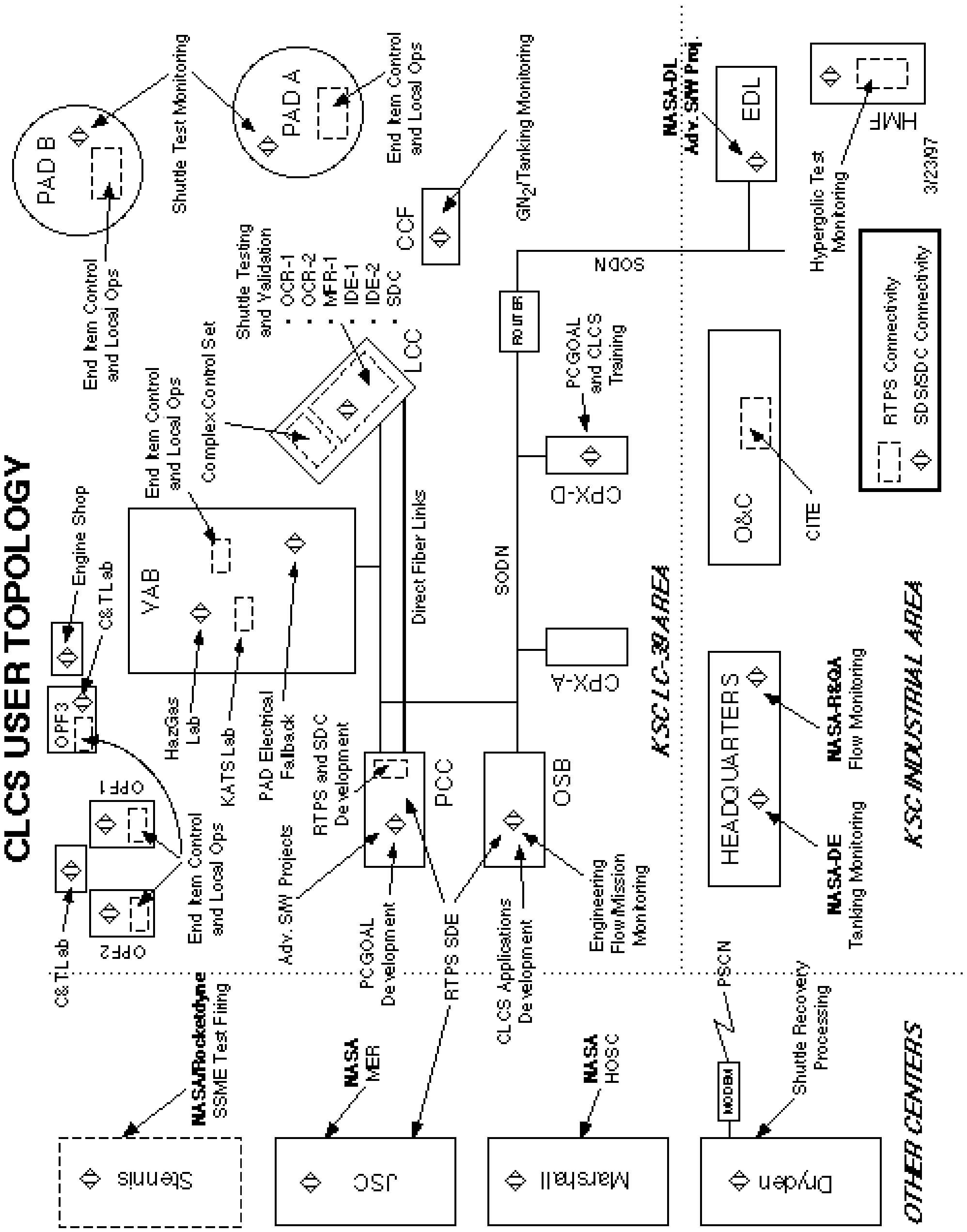


Figure 1-1 CLCS User Topology

1.3. Document Overview

This document is identified as the CLCS Concept of Operations. The document is organized into six sections. Section One contains the introduction of the operations concept and an operational overview of CLCS. Section Two contains definitions of CLCS physical characteristics and roles and responsibilities. Section Three is separated into two subsections, Support Operations and Test Operations, which describe how CLCS will be used for all aspects of operations. The fourth section discusses system processes, including hardware, system software, and application software. Section Five, User Environment, describes the human computer interface (HCI) capabilities necessary to support test operations in the control rooms and off-line processing sites. The final section contains operations scenarios describing integrated processing through launch and non-integrated multi-flow processing including simulation.

1.4. Applicable Documents

National Launch Processing System, Project Baseline (aka the "Blue Book")
Checkout and Launch Control System (CLCS) System Level Specification, April 15.1997, 84K00200
Checkout and Launch Control System (CLCS) System Design Document, Draft, 84K00210

2. Operations Concepts

This section will discuss the physical characteristics of the CLCS areas which includes the LCC Set, specialized processing sites, local operations, and the software development sustaining environment. The section will also discuss the roles and responsibilities of personnel supporting CLCS.

2.1. Physical Characteristics

2.1.1. Launch Control Center (LCC) Set

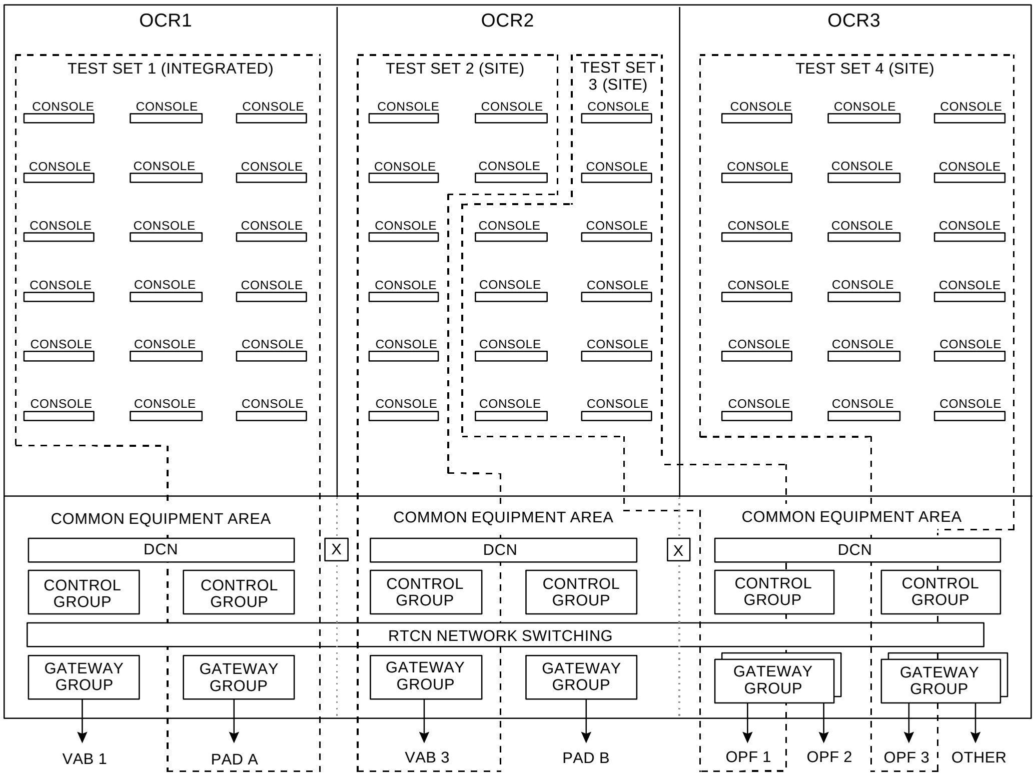
CLCS areas capable of providing command, control and monitoring of ground and vehicle processing for an entire mission, from landing to launch are defined as OCR's. The LCC set is comprised of three OCR areas. CLCS will have the flexibility to meet the changing demands of all aspects of Shuttle processing in a quick and reliable manner. The OCR will be configurable into flow zones to support multiple flows. A control room will typically be divided to support 2 non-integrated flows although the rooms can be divided to meet varied processing requirements. These configurations drive a variety of equipment allocations in the Common Equipment Area (CEA). The groupings of consoles in a flow zone along with their dedicated common equipment comprise a test set. Figure 2-1 illustrates an option of OCR test set configurations for normal day-to-day operations. From the start of integrated operations, defined as the period from the Shuttle Interface Test through Terminal Countdown and Launch or Abort/Safing and Scrub/Turnaround, an entire control room will be dedicated to support the prime vertical flow. For a portion of launch countdown and certain launch countdown simulations both prime OCR's will be configured to support the prime vertical flow. Figure 2-2 illustrates the planned launch countdown test set configuration. These test set figures do not accurately represent the floor layout of the control rooms. They are only intended to identify the concept of grouping common equipment with consoles to form a test set.

OCR-1 and OCR-2 will be configured to fully support integrated processing through launch. The third OCR will contain sufficient assets to process at least two flows. This room will be able to be reconfigured on short notice to accommodate daily test requirements. OCR-3 is not being designed to support major integrated tests or launch countdown since it will not have same support as OCR-1 and OCR-2 (e.g., certain hardwire safing links, etc.). Each OCR contains a shared input/output (I/O) area (not shown). Meant to replace the data review room, the shared I/O area allows the engineers to retrieve recorded data without leaving the control room.

The LCC Set is comprised of systems engineering type consoles, Test Conductor (TC) type consoles, the common equipment area, and management areas. Reference Figure 2-3 for hardware counts in each OCR. A test set is a subset of the LCC set used to support a flow. Each test set within the LCC Set will provide hardwire links to safe the vehicle (e.g. power down, propellant drain, etc.) and its environment (Pad, OPF etc.) completely independent of CLCS. The area will also provide dedicated printer support within headset cord range of all positions. Sufficient storage assets and table surface area will be provided for auxiliary data, books and equipment.

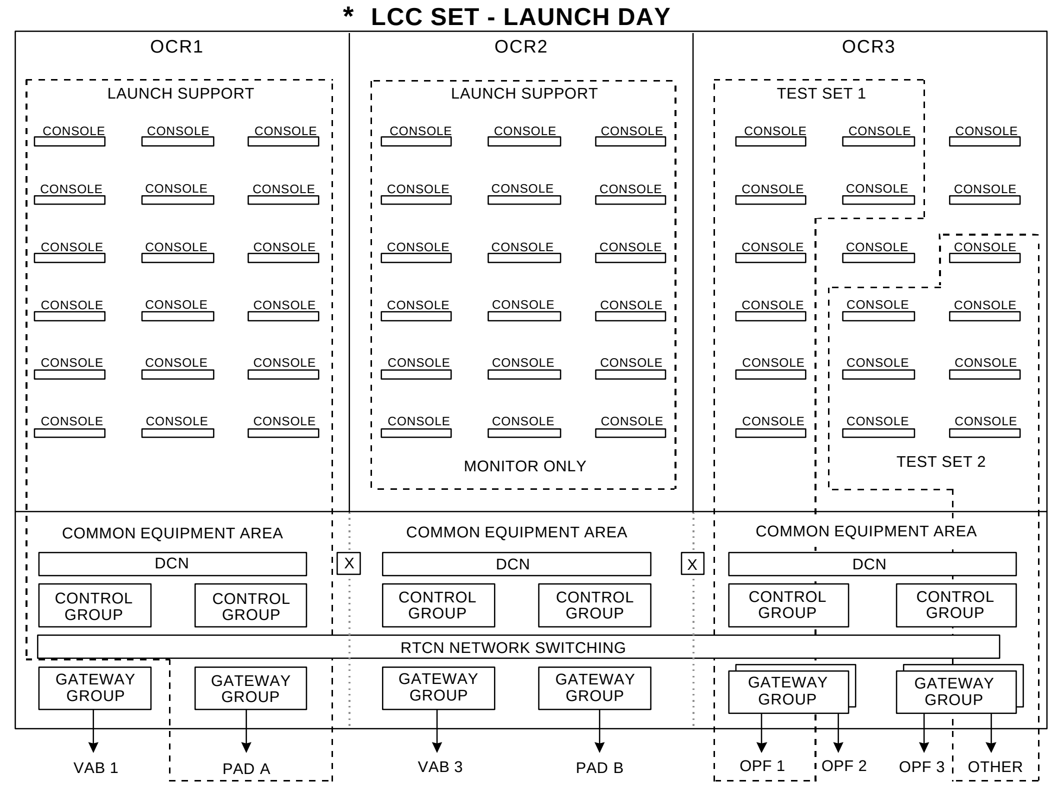
Furthermore, there will be other CLCS functionality in an OCR for critical facility systems control. Specifically, there will be control and monitoring of firex and sound suppression systems during launch countdown. Within the LCC set, facility control and monitoring would remain in its current location for non-critical testing.

*** LCC SET - DAILY OPERATIONS**



* Not representative of actual number or layout of consoles in OCR

Figure 2-1 Daily Operations Configuration



* Not representative of actual number or layout of consoles in OCR

Figure 2-2 Launch Day Configuration

	Test Conductor Workstations	OTV CRT's	Safing Panel	Phone	System Engineering Workstations	OIS-D Units/Chan (4 Ch)	Support Modules
OCR-1	18	97	42	78	42	60 (18 8-Ch)	21 (6 Cmd/Ctrl)
OCR-2	18	97	42	78	42	60 (18 8-Ch)	21 (6 Cmd/Ctrl)
OCR-3	12	69	6	68	36	36	18
OSR (x2)	0	5	0	9	0	9 (8 Ch)	9
OMR (x2)	0	5	0	9	0	9 (8 Ch)	9
CITE	1	N/A	N/A	7	6	6	4
SAIL		N/A	N/A	6	6	N/A	2
DRFC	2	N/A	N/A	5	3	N/A	6
Ice Team	1	28	N/A	6	0	8	6
CCS	1	14	1	3	2	3	2
HMF	0	5	1	5	5	5	4

The OTV, phone, and OIS quantities shown include the allocations for workstations and support modules.

Figure 2-3 Workstation Equipment Allocations

2.1.1.1. Flow Zone

The CLCS Flow Zone concept provides the capability to command, control and monitor multiple processing flows (or TCID's) within the same OCR. The control room is divisible into distinct areas for each TCID/processing flow. Command paths are logically isolated to those areas which support that flow. The distinct area will contain physical clues (i.e., signs, display headers, etc.) as to which flow a user is working on and can be configured on an as-needed basis to provide sufficient consoles for testing associated with that flow. Storage assets may be movable to facilitate area reconfiguration. The area will provide sufficient output assets (printers, faxes, copiers) to prevent interference between the systems testing within a flow and between Flow Zones.

2.1.1.2. System Engineer Consoles

In CLCS, the definition of a traditional "Console" changes. Consoles will be generic pieces of hardware that can support any engineering system testing on any given day. Consoles will no longer be associated with system discipline because all displays are resident at every location. Systems will still be consistently grouped within a control room on a day to day basis for normal test operations, whenever possible. There will be a standardized configuration identified for major test operations. The console will contain updated technology to display data and incorporates separate command, monitoring, and business information networks. There will be a dedicated area on the console for Operational Intercom System (OIS), safing panels, phones, and other institutional legacy type equipment. Figure 2-4 shows a conceptual system engineering console workstation.

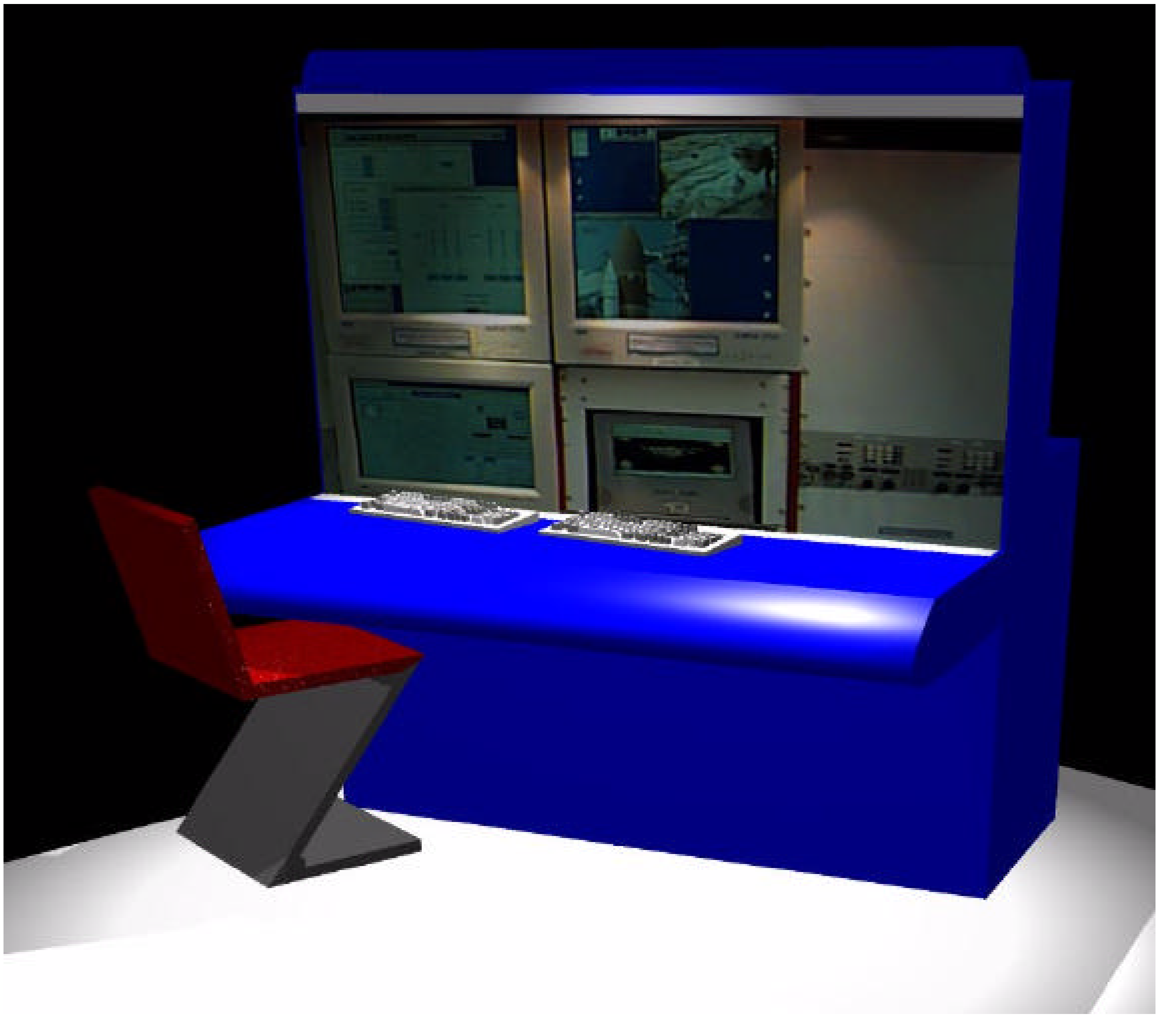


Figure 2-4 Conceptual System Engineering Console Workstation

The Human Computer Interface (HCI) at a console workstation associated with the command and control network will provide all users with standard, consistent interfaces. The interface is the same for all users (engineering, test conductor, and management).   This standardization is intended to reduce the training time and present a universal approach into the command and control network.

A “wrapper” will be developed for CLCS. The “wrapper” contains items as: Universal Time, vehicle, top level view of links, formats, stopwatch, commandability (see below), applicable displays and other standard information that is universally required by the user community. Specific information within the banner may change (e.g. links) depending on the test set (LCC, CITE, CCS, HMF). The overall banner, however, will be consistent for all users. Within the “wrapper”, standard interfaces will be provided to access application programs (displays, command and control screens, state machine status, etc.), system provided services (Plots, CLCS subsystem health, viewers etc.), and other information (including on-line help, text editors, etc.). Similar to the banner, specific information within the interfaces changes based on test set, but overall format and layout remain consistent.

Each console will be able to view any and all displays that are deployed in the particular test set. The difference between console workstations is the commandability associated with the station. Consoles will only be able to send commands if they have been granted specific authority. A list of user classes (somewhat similar to today's systems) will be developed and will have a defined range of commandability. These functions may be shared among several groups. During application software development, the user community will determine the commandability baseline. In the operational environment, the ability to temporarily add to or subtract from the commandable baseline will be provided. A method will be provided in the standard banner to show the command authority granted to each console workstation. The default capability will be NONE, which allows a console workstation to be in "display only" mode. As a minimum, commands that can be issued from the HCI (to an end item) will required a "two step" (arm/execute) type invocation action. The method will be standardized, where applicable, for all programs available through the HCI. Figure 2-5 outlines the functional network connectivity of the system engineering console workstation.

The system will provide monitor capability for management and support personnel in an operational environment for launch (i.e., as FR-2 is used today). These personnel will be able to view the same displays that are being viewed on the launch room command screens, but they will have no command capability.

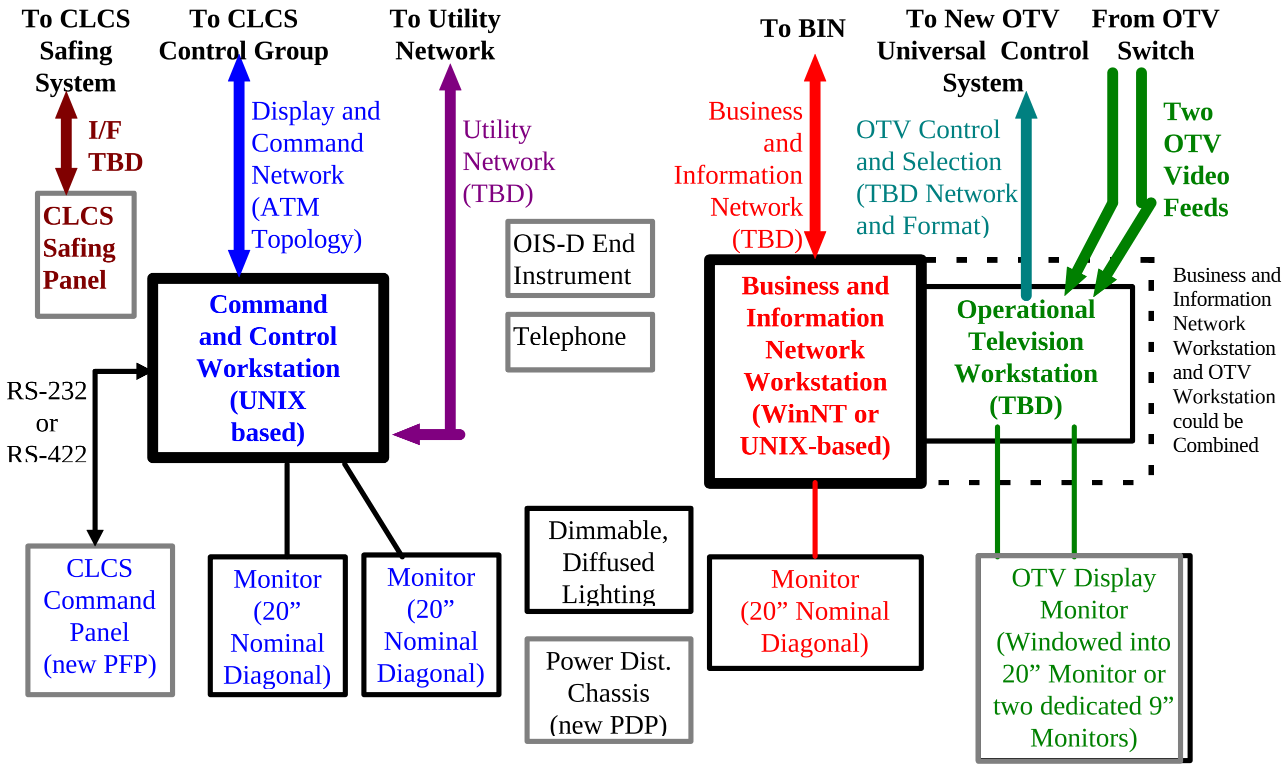

To CLCS Safing System
I/F TBD
CLCS Safing Panel
To CLCS Control Group
Display and Command Network (ATM Topology)
To Utility Network
Utility Network (TBD)
OIS-D End Instrument
Telephone
Command and Control Workstation (UNIX based)
RS-232 or RS-422
To BIN
Business and Information Network (TBD)
To New OTV Universal Control System
OTV Control and Selection (TBD Network and Format)
From OTV Switch
Two OTV Video Feeds
Business and Information Network Workstation (WinNT or UNIX-based)
Operational Television Workstation (TBD)
Business and Information Network Workstation and OTV Workstation could be Combined
CLCS Command Panel (new PFP)
Monitor (20" Nominal Diagonal)
Monitor (20" Nominal Diagonal)
Dimmable, Diffused Lighting
Power Dist. Chassis (new PDP)
Monitor (20" Nominal Diagonal)
OTV Display Monitor (Windowed into 20" Monitor or two dedicated 9" Monitors)

Figure 2-5 CLCS System Engineering Console Position Functional Connectivity

2.1.1.3. Test Director/Test Conductor Consoles

Test Directors and Test Conductors (TC)serve a purpose in the control room unique from that of systems engineers. Provisions for such things as expanded OIS and Operational Television (OTV) capability, RF OIS capability, Astronaut Communication panels, Paging & Area Warning Systems, Ground Launch Sequencer (GLS) hold switch, Emergency Camera Panel, Bird Deterrent Panel, and multiple phones will be required. There are no differences that represent system architectural drivers since these consoles require access to the same networks used by systems engineers.

There are, however, different configuration requirements for the TC consoles based on specific job functions. Some TC consoles need the capability to enable or disable the computer network for CLCS local operations and to alter control room configuration during day-to-day operations (i.e., increasing or decreasing the number of workstations in a flow zone). Each OCR will have two support modules on the TC rows to serve as command network access for the Test Conductor command authority management tasks. Network feeds for additional access to the command and control network will be provided to every other TC/TD position for flexibility in future operational plans. Figure 2-6 shows the TC console position functional connectivity.

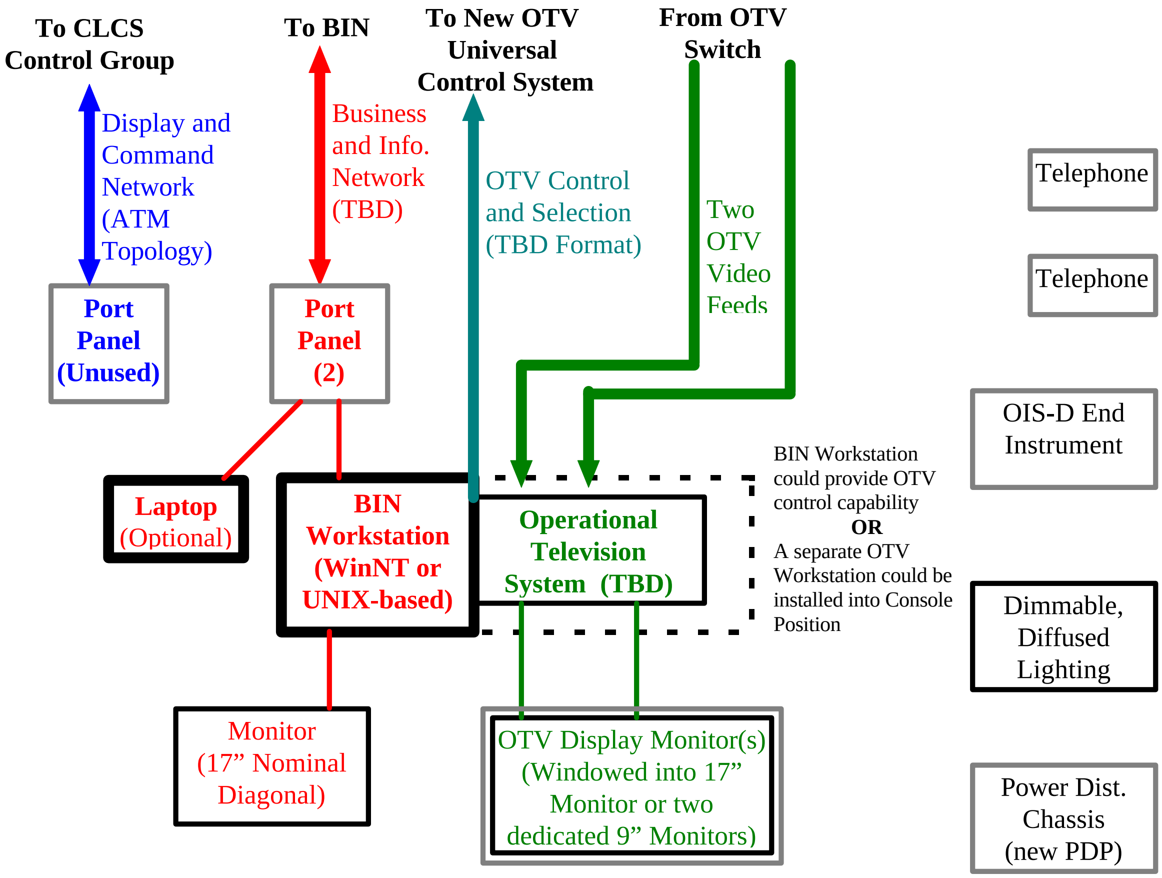

To CLCS Control Group
Display and Command Network (ATM Topology)
Port Panel (Unused)
To BIN
Business and Info. Network (TBD)
Port Panel (2)
To New OTV Universal Control System
OTV Control and Selection (TBD Format)
From OTV Switch
Two OTV Video Feeds
Laptop (Optional)
BIN Workstation (WinNT or UNIX-based)
Operational Television System (TBD)
BIN Workstation could provide OTV control capability
OR
A separate OTV Workstation could be installed into Console Position
Monitor (17" Nominal Diagonal)
OTV Display Monitor(s) (Windowed into 17" Monitor or two dedicated 9" Monitors)
Telephone
Telephone
OIS-D End Instrument
Dimmable, Diffused Lighting
Power Dist. Chassis (new PDP)

Figure 2-6 CLCS Test Conductor Console Position Functional Connectivity

2.1.1.4. Operations Management Room (OMR)/Operations Support Room (OSR) Areas

Program and Center management support of launch countdown from the OMR and OSR require a different “console” configuration than the rest of the control room. This area should have access to the real-time data and informational data networks along with OTV and OIS. The areas will be configured more as executive management areas with support modules in place of consoles. Each support module will have a BIN workstation, OIS and phones. Monitors will be placed around the room. Reference Figure 2-3 for equipment allocations.

2.1.2. Local Operation Definition

Hands on, face-to-face communication with on-site personnel is an important improvement to the control of operations. The ability of an engineer on the floor, at the vehicle/processing facility, to directly monitor and command operations and perform troubleshooting on-site is a useful tool. CLCS local operations are not satellite control rooms; in this context, the purpose is to carry the test to the vehicle which eliminates the need to have an engineer supporting the task from the OCR. It will allow closer coordination with the task team working the job.

CLCS local operations will include the ability to monitor and command flight and ground hardware from multiple places in all processing sites. This will be implemented by extending the required network(s) to all processing sites. Each site will have network drops installed near vehicle and ground processing locations. The number of drops will initially be limited to key locations such as the crew module, near the orbiter aft access door, at the cryogenic storage facilities, and the MLP’s. The concept is scaleable with additional drops installed to meet specific engineering requirements.

All network drops at a local site will be enabled for monitor only. If command capability is needed (limited to the activity authorized for that location), activation by the Test Conductor will be required. An activation protocol will be established between the local user and the TC.

Portable workstations that provide monitoring, command, and limited advisory systems access are required. Although these workstations will only need a limited set of the capabilities of an OCR workstation, they must provide the capability to control both single commands and command sequences. They will not be required to support OTV interfaces or access to the Business and Information Network (BIN). It is acknowledged there will be some delay in command transmission/response times, therefore, time critical commanding does not need to be supported. Reactive safing will only be supported through the OCR.

Use of local control for hazardous operations will be limited. Local control of operations will not be allowed should the loss of command capability increase the hazard, or if potential exists for a hazardous area to expand to include the control location. The intention of local operations will be to primarily support single system element testing. Testing requiring multiple system integration should remain in the OCR under the purview of a TC.

2.1.3. Specialized Processing Sites

Specialized Processing Sites are defined at those areas outside the LCC set where CLCS activities will be performed. These include the HMF, CITE, Shuttle Avionics Integration Laboratory (SAIL), DFRC, and CCS. These sites will require both Systems Engineer and TC type consoles to support testing requirements. CLCS will provide the same capabilities at these sites as identified for the OCR's.

The specialized processing sites will have sufficient test sets installed to support the test requirements for that site. The console workstations and common equipment will be baselined to identify equipment required for these sites.

2.1.4. Software Sustaining Environment

Software Development Environment (SDE) and Integrated Development Environment (IDE) are areas outside the OCR's where CLCS hardware and software are developed and tested prior to release for use in actual test operations. Four SDE and at least two IDE areas will be necessary to support hardware and software development and sustaining activities.

The SDE includes fixed locations and a desktop development environment. These areas will provide all capabilities necessary to emulate the CLCS system to facilitate testing. The four SDE areas are:

1. SDE-1: This set will be used for the development of system software and hardware in a test environment with informal configuration control. It will consist of development and operational workstations, networks, real-time processors, and gateways.
2. SDE-2: This set will be used for the development of system software and hardware in a test environment under formal configuration control. It will contain the same elements as SDE-1.
3. SDE-H: This set will be used for the development of system software in Houston in a test environment with informal configuration control. It will contain the same elements as SDE-1.
4. SDE-Users: This will be a set of development workstations in the PCC and OSB to permit the development of software and associated documents in an office environment. This environment is supported by Configuration Management servers located in the controlled environment of SDE-1, SDE-2, and the SDC.

In addition to the SDE areas, there will be a simulation development environment that will provide a desktop capability supported by SDE-Users. The simulation development environment will provide a stable execution system and configuration to develop and sustain math model applications.

The IDE will provide a near-operational system where software can be tested against a stable hardware and system software configuration. In order to obtain the maximum benefit of the IDE, it will have all development tools and languages available during validation and system checkout. The two IDE areas are:

1. IDE-1: This area will be a set of actual CLCS hardware in a test environment under formal configuration control. The set will be configured just like hardware deployed in an operational environment to support application validation and user acceptance.
2. IDE-2: This area will be a redundant set of CLCS hardware in a test environment under formal configuration control configured as a real set to allow total system checkout.

2.2. Roles and Responsibilities

2.2.1. Users

Users are personnel who utilize CLCS resources to support and perform processing test operations. Users are divided into three groups defined by the types of tasks they perform. The first task is test preparation. Users who support test preparation include Test Application Developers, Test Data Base Developers, and Test Configuration Builders/Managers. Preparation consists of those tasks which are required to prepare all aspect of test definition and configuration.

The second task is test execution. Users who perform test execution tasks include System Test Engineers and Test Conductors. Execution consists of control and supervision of CLCS test operations.

Users who perform Support Administration tasks include System Administrators, System Configuration Managers, and System Data Base Administrators. Administration consists of account maintenance, configuration management, and maintenance of system data bases.

2.2.2. Operators

Operators are defined as personnel who manage and maintain CLCS system resources in support of processing. Operators are divided into two groups defined by the types of tasks they perform. The first task is Support Operations. Personnel who perform Support Operations tasks include Set Managers, Flow Zone Managers, Support System Engineers, Support Operators, and Network Managers. Support Operations consist of management and control of CLCS system resources and preparation and monitoring of each CLCS TCID in support of testing.

The second task is Support Maintenance. Personnel who perform support maintenance tasks include Maintenance Engineers and Maintenance Operators. Maintenance consists of failure analysis, fault isolation, recovery, and preventative maintenance.

2.2.3. Sustainers (Developers)

Sustainers are the personnel who provide support to keep the hardware, system software, applications software and data bank/system build configured properly and to update those configurations as necessary.

3. CLCS Operations

This section will discuss both support and test operations. Support operations are those activities that provide the framework for test operations. Test operations are those activities which are directly responsible for controlling and monitoring processing with CLCS.

3.1. Support Operations

Support Operations include all of the activities and tasks to define and prepare for the execution of test operations and to provide auxiliary services, as required.

3.1.1. Test Definition

CLCS Data Bank. The Shuttle data files from JSC and GSE Data files from KSC are used by the DBSAFE (Data Bank Shuttle Function Executive) process to update the CLCS data bank prior to beginning the test definition phase. The data bank contains detailed descriptions of all FD's (Function Designators) and measurements to be monitored or controlled by the CLCS. The data bank resides in the SDC (Shuttle Data Center).

TCID Build**.** A TCID is a compilation of all software, tables, files, and parameters needed to configure a CLCS test set to support a specific test operation. These tables and parameter files are also used to configure required SDC and simulation processes. The TCID build process is performed on the SDC.

3.1.2. Set Configuration and Start Up

Startup tasks and procedures are performed in order to prepare a group of CLCS resources to support vehicle or GSE testing. Based on the outcome of the resource manager function, the set hardware is configured as a test set with control group, gateway group, and network segments. System integrity monitoring is enabled and the scheduled TCID software and tables are distributed, along with the associated operating system. The test set is loaded with the software and tables required for testing. Operational readiness is verified and system processes are initialized and started. Connectivity to the proper end item is verified and data acquisition and processing is started.

3.1.2.1. Set Configuration

Safeguards will exist to ensure proper set configuration. Hardware configurations will be checked and validated by software all the way to the end item for that test configuration. Software will be available to monitor the configuration at various levels including the LCC set, test set, Flow Zone, user positions, and individual components. This software should provide configuration information about the TCID's loaded in each subset, each user application set, overall resource sizing information, all network components, etc. All configuration information will be recorded.

3.1.2.2. Set Load

In order to reconfigure a set from one TCID to another in a reasonable time, multiple TCID storage and automatic load capabilities are needed. Several versions of system software and TCID's may be stored at the SDC. During the pre-load process, the Activity Manager function will retrieve the operating system software and the applications software for the appropriate test configuration from the SDC and transfer it to local application servers. Upon successful initialization of subsystems with system software, the application set required to support testing may be made automatically available to the user from the local applications server.

The loading of software to resources within a set is managed by the activity management task. When invoked by the master function, this task provides the capability to load an entire CLCS test set, a specified group of resources within a set, or an individual resource. The activity management task provides the capability to request a change (load or reload) to a test configuration (in real time) in response to a request from the master function. Additionally, it obtains integrity and version/revision information of software currently stored on a server or hardware platform, and can compare this information to pre-defined test configuration parameters. When an entire set or hardware elements within a set are to be loaded, a test configuration which has been predefined and stored in a centralized storage location within the SDC is loaded into the target hardware. The software to be loaded consists of the operating system, system applications and user applications, as well as supporting tables, etc., that make up the test configuration software package (TCID). The software test package is built by the software development process and placed under configuration control prior to being loaded into an operational environment. When a configuration is needed for testing, operations personnel retrieve the specific configuration and, upon verification of the desired configuration, initiate the loading of the specified hardware which will continue autonomously. The activity management task associates a group of gateways, processors, servers, etc., with a software test package as pre-defined by the test configuration, in addition to the group of user positions (workstations) required to support the test configuration. The activity management task monitors the progress of the load process, while the SDC loads the hardware and provides data integrity and verification mechanisms to insure proper subsystem load and configuration.

3.1.3. Operational Readiness Test Task

The Operational Readiness Test task is supported by the master function, which provides the capability to safely test system resources. These tests provide a "level of confidence" that a resource is properly configured and ready to support in an operational environment. This task exercises the software and hardware interfaces, to the greatest extent possible, which are utilized when the resource is functioning in an on-line (operational) environment. Specific restrictions are incorporated such that testing of resources that are directly connected to a end item do not adversely effect the end item itself.

3.1.4. Network Manager Function

The Network Manager Function is responsible for the management, configuration, and monitoring of all CLCS provided network components. A network manager function is provided for the LCC set to manage these tasks. It provides health and performance data to all master functions (for the system integrity and activity management tasks) within its OCR. Any network related problems are first detected in the Network Manager Function then passed to the master function to allow operational reconfigurations and to the maintenance monitor function for analysis. As it relates to the organizational level maintenance activities, the network manager function provides performance and configuration data to the maintenance monitor function for use in load balancing, system tuning, and both proactive and reactive fault isolation and repair. It also provides the control mechanisms for permitting or limiting access by all external network interfaces. Dedicated hardware monitoring test/injection points strategically located throughout the CLCS hardware sets support the connection and use of commercial products as appropriate to the specific network medium. The network manager function, working in conjunction with these commercial tools and other subsystem components, provides an integrated platform from which to base all control, reconfiguration and repair actions.

3.1.5. Data Retrieval and Distribution

SDC will provide the capability to retrieve data which has been recorded. There are two types of recorded data: packet data and distributed data Packet data is temporarily available on-line after which it will be saved on disk. Distributed data will also be available on-line based on disk storage capacity but will then be archived to disk. SDC will also distribute the same distributed data (not the Real Time Processing System (RTPS) packets themselves) as a consolidated data stream.

Existing retrieval applications which are applicable to CLCS will be used. SDC will provide at least two basic retrievals specifically designed for CLCS data. For packet-type data, SDC will provide a retrieval capability to select a time-span, packet type, and then receive a formatted dump. For distributed data, SDC will provide a retrieval to report all aspects known about the data including time, value, and health for the selected function designators.

3.1.6. Simulation

The Simulation System provides support for the testing and validation of CLCS equipment; checkout and validation of software used in Shuttle ground testing and launch operations; training of CLCS console operators; and launch team training. To support these activities, a number of master math models will be built to provide integrated simulation support. Master math models will include all planned and identified math model updates to support the respective TCID. The master math models, both debug and verified, will be groupings of system models to provide the simulation support for related systems and activities. Debug master math models will provide support for application software development and for training simulations (countdown, cryogenic loading, and hypergolic loading).

Each new TCID build or edit will require a build or rebuild, respectively, of the master math model for simulation support. The master math model will be built by the software development process prior to being loaded into an operational environment. When a configuration is required for simulation support, operations personnel will retrieve the specific configuration and, upon verification of the desired configuration, initiate the loading of the specified hardware which will continue autonomously.

3.1.7. Operational Maintenance

The CLCS maintenance methodology is a broad-based approach to operational level maintenance which utilizes system performance and failure data, on-line and off-line diagnostics, and powerful software tools to meet the requirement to isolate 90 percent of all non-intermittent faults to the Line Replaceable Unit (LRU) level. Generally, an LRU is a circuit card, power supply, network or computer box, printer or other assembly whose repair is not practical within the on-line environment.

3.1.7.1. Maintenance Monitor Function

The Maintenance Monitor Function is key to the overall success of the on-line environment in meeting the identified Mean-Time-To-Repair goal of 30 minutes. It is a multi-faceted tool, dedicated for maintenance use, which provides several capabilities. It serves as a central point to run both intrusive and non-intrusive diagnostic testing on all subsystems associated to any OCR. Intrusive testing is allowed only on subsystems which have been released from operational support. This diagnostic functionality includes the ability to be used in a debug/monitor mode to capture and log system boot up processes for use in diagnosing subsystem failures. It collects time-tagged operational failure and health data from the network manager and master functions such as system messages, health and status, network statistics, and Operational Readiness Test (ORT) results. It also has access to historical databases currently being used as well as other reference data sources such as the Technical Notification System, and the Maintenance Planning Database. These data sources are used in conjunction with the operational and diagnostic failure data, and analysis software, to provide a comprehensive fault isolation tree to guide maintenance personnel through the troubleshooting process. The end result of this function is an ordered list of the most likely failing LRU(s). This expert/predictive software is also used to provide advisory recommendations to the appropriate master function concerning components that have not yet failed completely but are operating in reduced or degraded fashion. This ability allows for timely operational reconfigurations of failing hardware. This high level of integration of operational and diagnostic data sources helps manage maintenance actions to provide the highest level of operational availability to the user.

3.1.7.2. Local Diagnostics

For those systems which are unable to communicate with the maintenance monitor function due to primary network related problems or other faults, local diagnostics which reside on each subsystem are used to isolate faults. These diagnostics are a collection of vendor and custom designed tools, utilities and fault isolation processes. Disk-based, time-stamped error logs and vendor supplied power-on self test functions, residing on each subsystem, are used to provide additional insight. Further, connectivity is provided from the console port of each subsystem to the maintenance monitor function to capture boot up failure processes and other messages which may be routed to the port. This ensures that valuable failure data, which would otherwise be lost, is preserved for diagnosis/evaluation.

3.2. Test Operations

Test Operations will include all activities and tasks to perform actual processing of the ground and flight elements.

3.2.1. Test Control

Consoles supporting a specific TCID will be grouped in a designated area within the control rooms to allow maximum flexibility in the use of hardware and personnel while reducing the number of idle consoles.

The Test Set will be configured, loaded and initialized by the master function prior to test engineer and vehicle test conductor access. Command and control consoles will allow only display capability until command assignments are established.

3.2.2. Command Authority Management

Access to the command and control network in CLCS is generally restricted. The initial state of consoles in CLCS is display only (i.e. commandability = NONE). Users must coordinate through the authorizing agent (i.e., test conductor, master, integration) to be granted the desired command authority. A resource allocation function will be provided which grants (and removes) command authority to console positions.

Users are not required to possess a user ID to access the command and control network (an exception is detailed in the next paragraph). Command authority is granted to a console position where the user works. In CLCS command authority is granted after request by the user.

The only user class that requires a logon and password is a subgroup of the master function. A controlled access method is required because they are the enablers of the test set. Once the set has been enabled, other personnel could perform the load and initialization functions. The User ID in this case is not used to identify the person performing the work, it merely controls “root” access.

Existing user OIS discipline will be used. User will coordinate commandability requests (changes) through the test conductor.

A mechanism within the resource allocation function will be available to configure the control area in a predefined fashion (i.e. standard power up, Launch Countdown, Hyper-load, etc.). There are no restrictions on the relationships between console position and commandability availability. Multiple consoles may be assigned the same user class (e.g. there may be as many LOX commandable consoles as required). A single console can be granted commandability for as many systems as required by the user (there could be DPS, INS EPD, ECL and ECS command capability at one console).

During critical operations provisions should be available to quickly move command authority should a workstation fail. With a predefined layout, this reassignment precludes a loss of system control. For example: A failed console's commandability could be assigned to an adjacent workstation with the operator moving only a few feet. This transaction would be virtually instantaneous.

In the command and control network the lowest level of workspace is the group. Personal files and unique startup files do not lend themselves to a team approach. If there is information important enough to be stored, it must be available to all members of the group. Individual file space is not required or desired.

Messages (system and application) are processed and recorded continuously. System messages cease when the system is terminated. Application messages stop when the Application terminates (regardless of user access). A message viewer is provided to allow the user to view any desired message type (or all).

3.2.3. Application Start and System Status

After successful command authentication, the user will have access to all applicable applications available to that console position to perform a specific task. Any combination of windows, test applications, etc., can be opened. Program initialization commands may be initiated by the user via keyboard entry at the dedicated command line area or from a graphical selection mechanism using a keyboard or a pointing device.

3.2.4. Command/Monitor Application

Commands may be initiated by the user via keyboard entry at the dedicated command line area or from a graphical selection mechanism using a keyboard or a pointing device. The system software then can perform all the related syntax and semantic checking regardless of the method utilized by the user to enter the command. After application programs are brought up for the appropriate subsystem/task, user will monitor test operations and respond to system messages, change configuration if necessary, report and solve problems, and terminate test.

3.2.5. Consolidation Efforts

It is intended that advancements in application software and displays will allow significant consolidation in day-to-day multi-flow processing. One such instance is the babysitting concept for vehicle power.

Since like systems will be grouped together, standard vehicle power up/down of the INTG, ECLSS, EPDC, ISL and DPS systems and monitor and control of vehicle power will be able to be performed from a single console position. This orbiter multiple system monitor position will be capable of monitoring for other systems, recovering from component failures, performing routine vehicle operations, and executing an emergency vehicle power down. A console position of this type will be available for each flow zone. System unique testing will still require dedicated console support.

Furthermore, the ability to perform multiple flow functions (e.g., multiple vehicle power up, babysitting, etc., from OCR-3) shall be designed into the CLCS architecture.

3.2.6. Data Recording

The SDC provides a centralized recording capability for time-tagged Shuttle processing data, RTPS configuration changes, error messages, system messages, inter-process communications, and other information. The data is recorded in two different ways: packet recording and data recording.

Packet recording provides the capability to record packet-like traffic which occurs on the RTCN and the DCN. This data is recorded with a Universal time-stamp. Data recording breaks apart the packets of data into the individual components (e.g. FD data, messages, etc.). The SDC then sorts, indexes and organizes this data for efficient retrieval. This data is recorded using Data Stream time. When packets from the various subsystems within a test set are put out on a RTPS network, the packets are time-stamped (by the Gateway) that is being acted upon by that subsystem at any given time. Thus, in order to properly correlate events relating to distributed data, these packets must also be time-stamped using Data Stream time. In support of this, however, the actual Universal Time stamp when the packet containing the data was placed on the network is also available. A benefit of the Universal Time embedded with the recorded data is the ability to easily correlate the recorded data with the recorded packets.

3.2.7. Data Access

Real-time data network access will be available to all areas; including OCR's, OSB, local sites, and processing sites. There should be no restrictions on shipping data to any NASA or Contractor facility. CLCS will have a capability to restrict displayed and recorded data on the network during impoundment periods caused by incidents or contingencies. Access to the command network will be restricted to control rooms and dedicated LC39 local sites only. The business system interface that will be provided must be isolated from the command and real-time data networks. This interface may differ from site to site.

3.2.8. Operations Redundancy Definition

For the purpose of this plan the word hazardous means that the loss of CLCS during a test would increase the risk of injury to personnel or damage to hardware.

As a minimum, the entire system will provide fail safe capability. Specifically, all console workstations must be fail safe. Primary critical common equipment, such as gateways, processors, and network components must be fail operational or have fail over capability. This redundancy must be provided for critical command and data paths before it can be used to support hazardous operations. The only exception to this will be the user display system in the console since adjacent positions will have independent power and data feeds. System engineering will require notification in a means other than through OIS when the console or application has failed or stopped responding (e.g., the redundant position is notified through CLCS of the primary position failure).

3.2.9. System Integrity Task

The System Integrity task performs the health and status monitoring of all resources (subsystem and network resources) allocated to a particular test configuration, and performs the redundancy management of all active/standby subsystems. It validates subsystem health, and based on system configuration, commands active/standby switchovers for failed redundant components. The system integrity task sends messages to the workstation(s) associated to the master function for notification and displaying of failed subsystems, or significant events that occur within the set. Selective messages are also sent to all subsystems in the affected set, to the SDC for recording, and to the maintenance monitor function for performance/fault analysis. Additionally, the system integrity task reports changes to subsystems (GSE bus errors, PCM format changes, loss of PCM data, etc.). Selective informational and error messages that are received from subsystem integrity tasks within all set resources, are processed by the system integrity task and are also routed to the Maintenance Monitor Function for fault analysis. Redundancy also exists for the System Integrity task. A graphical user interface provides the capability to monitor both overall system health and status, or detailed subsystem health and status. The network or subsystem status display is selected by the operator to the desired level of detail. For example, the operator can select to see a subsystem’s peripheral error counts, or the overall subsystem status. The health and status of each subsystem is also recorded in the SDC.

3.2.10. Subsystem Integrity Task

A Subsystem Integrity task is incorporated within all subsystems, and interfaces with the system integrity task. The subsystem integrity task provides health and status information to the System Integrity task for health monitoring and redundancy management decisions, and also records the health and status data to the SDC. Additionally, the subsystem integrity task routes error and informational messages to all workstations associated with that subsystem and to the associated master functions within the set. Each message is tagged as to the set and subsystem that the message originated from, and the time and date of the message occurrence. The message also contains an identifier of the software component originating the message.

3.2.11. Checkpoint Task

The Checkpoint task tracks and/or restores subsystem state information to a subsystem. The Checkpoint task will provide ongoing (continuous) and snapshot (demand) state information request capabilities, along with interfaces to other tasks that are utilized to restore a subsystem to a specified state configuration.

4. System Processes

4.1. Hardware

Hardware processes consists of the tasks pertaining to problem identification, resolution and upgrade of the Test Set equipment.

4.1.1. Hardware Maintenance

At the system level, major assemblies are repaired by removing and replacing LRUs with functional spares. Repair disposition will be determined by the Maintenance Plan. After receiving Commercial Off-the-Shelf (COTS) hardware from the vendor, and after repairing custom hardware, Intermediate Level Maintenance Facility (ILMF) performs a thorough retest/validation of all LRUs prior to returning them to stock.

4.1.1.1. Organizational Level Maintenance

Maintenance at the organizational level is generally limited to the periodic servicing of equipment through scheduled downtimes, troubleshooting to isolate failed components, and the removal and replacement of LRU’s and subsystem assemblies. Other actions may include in-place modifications, calibrations, line validations, and various reverification processes. These actions will be accomplished by SFOC contractor maintenance personnel. Any required vendor support within the CLCS program will occur in off-line environments only.

The emphasis at the organizational level is to return any failed hardware to an operational state as soon as possible to meet the operational availability requirement of 99 percent. The concept of Hot Spares is one of the methods used to meet this goal. The Minimum Peripheral Test Set (MPTS) is a hardware set located within the Launch Control Center which provides a complement of critical spares which are always in use, running diagnostics, or otherwise being exercised. These Hot Spares will be immediately available for installation in a control room during critical tests. For less critical support times, Organizational Maintenance relies upon off-line and on-line diagnostic tools, network health and statistics from the network manager function, and operational data from the master functions. These data sources supply operational health, status and test result information to the maintenance monitor function.

4.1.1.2. Intermediate Level Maintenance

The Intermediate Level Maintenance Facility (ILMF) provides intermediate and selective depot level maintenance for CLCS equipment using systems and procedures as described in each system’s Maintenance Plan. The Maintenance Planning Control Center (MPCC) within the ILMF is the primary hardware disposition area for CLCS hardware and manages the Return-To-Vendor processes. Hardware assemblies, repaired by and received from vendors, are verified operational and compatible with existing systems prior to assignment as serviceable spares. Independent verification and validation tasks are performed at the ILMF in a simulated on-line system environment to insure system compatibility of multi-revision level COTS, LRUs and Shop Repairable/Replaceable Units (SRUs). All assemblies are verified prior to use. No LRU or system components will be used in the on-line environment unless they have been verified. This reverification process may occur within the MPTS for large subsystems or for those subsystems which are impractical to transport to the ILMF. Automated Test Equipment (ATE) is used to support the repair of any custom and selected COTS hardware, where practical.

4.1.1.3. Depot Level Maintenance

Generally, Depot Level Maintenance will be performed by the vendor at the vendor facility. However, there may be circumstances in which these repair actions may be better accomplished at the ILMF. The Return-To-Vendor or Repair-On-Site decisions and considerations associated with COTS products are based on the most cost effective method to maintain these products and include:

- warranty considerations
- proprietary components and/or documentation
- test methods and procedures
- revision level management processes
- limitations of on-site resources such as tools, test equipment and skills
- maintenance agreements

4.2. System Software

System Software consists of the tasks pertaining to upkeep, upgrade and configuration control of system software and system supplied services/tools software. This includes COTS product upgrade/installation and integration.

4.2.1. System Build & Load

System build is identified as the definition, build, and load of target CLCS Sets that are independent of test build. A single system build can support multiple test builds and test build revisions. The definition and build portions are independent of the hardware destination CLCS set. A CLCS Set class defines the scaleable architecture makeup of the set but does not limit the quantity of subsystems in a target CLCS Set.

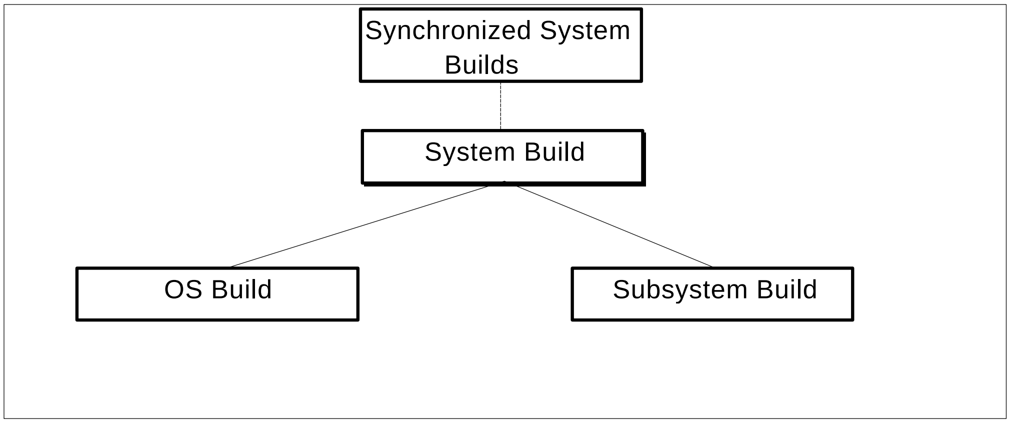


Figure 4-1 System Build Process

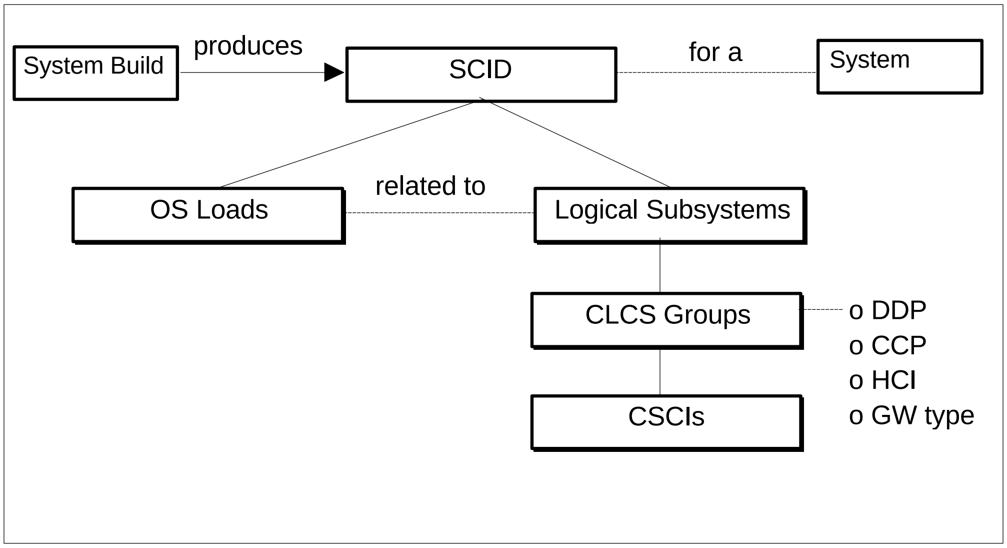


Figure 4-2 Set Build Configuration (SCID) Organization

4.2.2. Data Bank-System Build

Data Bank-System Build consists of tasks involving requirements capture, source product identification, engineering assessment, build execution, test/verification execution and release.

4.3. Application Software

The application software development task will include requirements capture, engineering assessment, software development (or modification), software validation and user acceptance. Application software will be developed using COTS tools where possible. There will be three types of application software supported in CLCS: simulation software; command, control, and monitor (CCM) application software; and test build and load software.

4.3.1. Simulation Software

The process for developing Simulation Software in the sustaining engineering environment includes all of the traditional aspects of software development, but utilizes them in an iterative and non-serial development cycle. The start of a follow-on phase is not dependent upon the total completion of its predecessor phase.

1. Requirements Definition: This phase involves the analysis of the initiating document (e.g., PRCBD, SSCR, RCN, ESR) to define the functional requirements to be supported by the simulation software.

2. Engineering Assessment: In this phase, simulation software design changes required to implement the functional requirements are defined and the cost associated with the change is estimated.

3. Software Development: In this phase, math modelers develop software to emulate the functional requirements of the hardware modification or replacement. All simulation software development will be performed using the simulation system software development tools and development environment.

4. Software Debug: Simulation software debug will be performed on all developed and/or modified simulation software prior to being configured to a validation environment. This testing will be performed utilizing a desktop testing environment. Final simulation software debug may be required to occur in the OCR to ensure all system related issues have been addressed. During the software debug phase, the simulation system will support the capability of modifying and loading a new software revision in a minimum time frame.

5. Software Validation: Upon successful completion of the debug test effort, the simulation software is validated by the modelers, and users when required, on the simulation system, or if necessary, in the OCR. This testing ensures the simulation software satisfies the functional requirements.

4.3.2. Command, Control and Monitor (CCM) Application Software

The CCM application software is planned to be separated into functional components such as, display monitors, end item managers, data fusion algorithms, constraints checking, data path health, and test application scripts. All application software functional components with the exception of the constraint check algorithm will be able to be updated from the CM system without deactivating the RTPS. CCM Application Software is categorized into several elements:

1. Display Monitors: These displays will be used to view information about the vehicle/GSE system components and to request, initiate, and control both commands and automated sequences.

2. End Item Managers (EIM): EIM will perform all direct FD commanding of vehicle/GSE system components and all automated sequences/tasks. The user will view the status of EIM's and sequencers through the finite state machine viewer.

3. Data Fusion: Fusion will allow combining values of multiple FD's and/or computations to derive a single value that can be used throughout CLCS. The user will specify the algorithm and elements to use for the fusion calculations.

4. Data Path Health: This will provide a mechanism for determining the health of individual FD's. This health will include all data path elements of CLCS (e.g., HIM's, gateways, etc.) as well as the vehicle/GSE specific path elements (e.g., power supplies, signal conditioners, MDM's). The user will specify the algorithm to use for the data path health determination.

The process for developing CCM Application Software in the sustaining engineering environment includes all of the traditional aspects of software development, but utilizes them in an iterative, non-serial development cycle. The start of a follow-on phase is not dependent upon the total completion of its predecessor phase.

1. Requirements Definition - This phase involves analysis of the initiating document (e.g. PRCBD, SSCR, RCN, ESR) to define the software functional requirements necessary to answer the request.

2. Engineering Assessment or Design - This phase maps the functional requirements into the Application Software Set architecture and existing design. During this phase the design changes necessary for implementing the functional requirements are defined and the cost associated with the change is estimated.

3. Software Development - This phase takes the software design and translates it into code using the CLCS defined languages and development tools. Re-use of software components will be employed (where practical) within and across Application Software Sets to help reduce the level of code that must be maintained. All development will be performed off-line from the OCR operations.

   - Display Monitor development will employ a COTS tool for creation/modification of dynamic data visualization displays. This provides a rapid development capability that does not require extensive code development

   - End Item Manger development will employ COTS languages and tools. All Application Software Sets will use the same development tools to ensure compatibility and commonality across the sets.

   - Data Fusion and Data Path Health elements will be developed/modified using System Software provided editors.

   - Debug Testing will be performed on all developed/modified software prior to being configured to a validation environment. This testing will be performed utilizing a desktop test environment that emulates the CLCS system (including DDP, CCP and all of their inherent functions). Upon completion of testing in this environment, final debug testing will be conducted in the Integrated Development Environment (IDE) to ensure all system related issues have been properly addressed. During this testing phase, the system will support the capability of modifying and loading a new software revision in a minimum time-frame.

4. Software Validation - Upon successful completion of the debug test effort, the software is validated by the users in the IDE, which provides a near-operational environment. This testing ensures the software satisfies the functional requirements.

5. User Acceptance - An operational environment connected to the simulation system will be used to allow user acceptance testing.

6. Test Procedure Modification - Where necessary, the test procedures that use the newly developed software will be updated to reflect any change in user interfacing or control mechanisms.

4.3.3. Test Build and Load

The test build process consists of the functions necessary to control and maintain the software required to support processing activities. The test build and load process is comprised of DBSAFE Software and TCID Build Software. Test build and load services will be integrated with the CLCS development environment and the SDC.

The test build function will determine whether to produce memory image tables or an interim form from which memory image tables are created at CLCS Subsystem load/initialization time. It will provide gateway tables, the online data bank table, the FD directory table, user application files organized by responsible system, and provide an initial capability to load and activate a Test Build onto a CLCS Set. A single test build can support multiple synchronized system builds.

The DBSAFE Data Bank provides a repository which describes all of the FD's used to support test activities. These FD's are the measurement and stimulus data used to control each individual end item (e.g. valves, switches, and gages on the vehicle, payload, and GSE). This data bank also contains the definition of user generated FD's (e.g. Fusion FD's).

Since the DBSAFE Data Bank supports measurements and stimulus required, the TCID Build software extracts the subset of FD data required to support a specific test or set of tests (e.g. the FD's relating to the specific tail number of the orbiter support, the ground facilities being used, and the specific FD's for the payload being launched).

This extracted set of FD's is then used to create the necessary files for the CLCS Target Set. The files created are the Online Data Bank which defines for the CCP, DDP, and HCI the function designator data applicable for the test. It also creates the Gateway Tables necessary for gateway functionality.

5. User Environment

CLCS will replace the basic capability that CCMS provides today, which is to perform test and checkout of the Space Shuttle. CLCS will enhance information gathering, display, and analysis capability through the implementation of new concepts and features that will be demonstrated by CLCS in its early deliveries. HCI Information Technology will be integrated into CLCS to provide design solutions for the needs of CLCS.

5.1. Safing

Currently there are four different types of safing in use in the control rooms. They are: Launch Data Bus (LDB) safing, GSE hardwire safing, program safing through programmable function panels (PFP), and reactive control logic. Provisions must be made to incorporate the functionality of each type of current safing in CLCS. New safing systems to support CLCS must be simple, reliable, and support the "generic console" concept.

5.1.1. LDB Safing

The CLCS architecture will significantly reduce the need for LDB safing by eliminating failures common to the current LPS architecture (i.e., console crash or buffer failure). However, an LDB safing equivalent is required for CLCS to protect against command server and command path failures. The preparation and maintenance of the LDB safing software will be part of support operations. The execution of the LDB safing software will be part of test operations.

5.1.2. Hardwire Safing

In CCMS, the failure of LPS can cause a loss of GSE or vehicle control capability for critical functions. In the CLCS environment, hardwire safing panels in the OCR's must be provided to ensure a backup control system for emergency safing and securing if a significant loss of CLCS occurs. Hardwire safing provides appropriate emergency responses and safing actions through a system wholly independent of CLCS. The hardwire system will also have a mechanism for continuously verifying its integrity. Sufficient hardwire links will be provided to support both integrated and non-integrated processing from the OCR's.

Hardwire safing will be implemented in a manner which allows consoles to remain generic. Hardwire safing from CLCS will be connected to the old hardwire safing system running from the LCC to the site. The power for the hardwire safing system must be separate from CLCS and have its own redundancy.

5.1.3. Programmable Function Panel (PFP)

In CCMS, PFP panels are used for sequencing tasks and executing safing sequences. Typically, Ground Launch Sequencer (GLS) safing and other critical safing sequences are activated with a PFP keystroke. PFP functionality is necessary but will only be used for critical safing operations. CLCS will provide PFP type capability but not necessarily in the same form as today.

5.1.4. Reactive Control Logic

In CCMS, Reactive Control Logic (RCL) is used to automatically perform safing/securing operations on predefined out of configuration situations that occur during testing. The functionality of reactive control logic must be preserved under the CLCS architecture.

5.2. Utility Tools

Utility tools include editors, viewers, and other specialized tools that implement capabilities which will be used both before tests are run as well as during test execution. For each of the functions described, the engineer will be able to create, edit, and view the associated data both in a developmental and an operational environment although there will be restrictions on these capabilities during processing activities.

5.2.1. End Item Manager Function

The End Item Manager (EIM) will be the primary interface for commanding a hardware or software component(s). The EIM will include closed loop and state-based control capabilities and be able to perform automated sequences of tasks. In addition to handling normal FD commands, the EIM will handle the equivalent of RCL and constraint management parameters for exception notification. An EIM may also provide a health indication to the CLCS system so its status can be monitored by the CCP manager function and the display manager (DM).

5.2.2. Data Fusion Function

Data Fusion involves computations using constants, measurement values, health values or other fusion values. Each fusion FD found in the data bank has the same attributes that any other FD of the same type would have with the exception that a fusion FD does not have a hardware record but does have a fusion algorithm table associated with it. Data fusion FDs are created through equations using combinations of FDs and various algorithms.

The Data Fusion Editor and Viewer Tool will allow the engineer to create, edit and view data fusion FDs. It will provide a GUI so that a user can easily enter and view information regarding a data fusion FD. The create and edit functions will be used during application software development. The view function will be used real-time in the control room to view not only the parent FD but the computations used to create that FD.

Data fusion will provide the capability to bring off-line data analysis on-line to the operator real-time as well as reducing software complexity by reducing the need for multiple programs to calculate the same data. Any FD may be used by any process throughout CLCS.

5.2.3. Data Path Health Function

Data Health is based upon a number of parameters, some of which may be external to the system. Data Health is the term applied to the integrity of a FD in CLCS. It consists of a group of flags which are associated with every FD. For example certain flags that deal with the decommutation of the data from its source are computed by processes in the CLCS gateway. Other flags correlate various data to determine additional “health” information about the FD’s. This information is available to all CLCS processes which utilize FD data.

5.2.4. FD Constraint Check Function

This tool will allow the engineer to create, edit and view FD constraints. It will provide a GUI so that a user can easily enter and view information regarding FD redline limits and messages. The create and edit function will probably take place in the office well ahead of vehicle testing. The view function will be used real-time to view the state of the system.

5.2.5. Command Entry

A command entry tool will provide engineers with an ability to enter “command line type” commands to an EIM. This tool will employ a GUI. The GUI’s menu will allow an engineer to choose a system and be presented with a list of applicable FDs. This list can be used to fill out the command entry form. Various safeguards will be implemented in this tool to avoid unsafe command entry.

5.3. Input Devices

5.3.1. Command/Business Network Separation

The command and control CPUs will be isolated from the business and information CPUs. This isolation enables the user to have access to a broad range of business and information computer systems without degrading the command and control system’s performance or security.

In order to completely separate access to the command and business networks, CLCS will provide keyboards for both networks. Each keyboard may be attached to more than one monitor but can only be connected to one network.

5.3.2. Pointing Devices

A GUI pointing device will be employed for CLCS. If there is more than one monitor connected to either the command or business network, the pointing device should be able to move between screens to support activities on both. Other mechanisms for pointing devices will be investigated by the design team as pointing device technology evolves.   The pointing device may be used to initiate access to menus and other interfaces, or optionally used to send commands to flight hardware or critical GSE in a two step fashion. Keyboard arrow keys should provide redundant control for cursor movement on critical command displays.

5.4. Output Devices

Auxiliary ports will be provided at each console with access to power, Business and Information Network, and peripheral devices. SDC data retrieval capability will be provided from the command network as well as the BIN at CLCS consoles.

The Common I/O area or Data Review area within each OCR should have two combination scanner/fax machines and three laser printers as a minimum.

Each console should have one laser printer and there should be one fax for every two console workstations.

5.5. Display Characteristics/Capabilities

The system should only require one set of displays to be built for all command and monitor functions to keep from maintaining more than one display for each function. For example, in CCMS, the command and control displays, PC-GOAL displays, and ESA displays are all different. In CLCS, these displays will be one and the same. All displays should be available on any monitor.

CLCS will provide a GUI to provide consistent access to the command and control network processes.. This interface will use standard GUI objects such as slider bars, pull down menus, popup menus and dialog boxes. This interface will be easy to learn and work with. A goal is to isolate the user from the operating system. The CLCS GUI will provide a task bar to enable users to quickly identify applicable processes. An activity indication will be provided for displays that are active, but not visible.

5.5.1. Message Window

The user will have access to a scrolled message window containing program and system messages. The user will have the ability to filter out messages from unwanted applications to provide operator focus on the primary task being performed. This window will provide visibility into the activity of the system.

5.5.2. Overlaying of Windows

Techniques for appropriate window overlaying at a CLCS command position will be developed, and corresponding software implementation standards will be instituted.

5.6. Business Systems

CLCS business and information CPUs should provide access to various management information systems supporting Shuttle processing and other NASA activities. Business and information CPUs, when switched to an appropriate keyboard and monitor, should become a client configuration for BIN applications.

BIN client configurations should be scaleable to support processor, memory, and disk storage upgrades as technology evolution in the data processing industry escalates the minimum configuration required to run mainstream office and productivity applications. BIN client configurations should also provide access to intracenter and intercenter mail and messaging capabilities through interfaces x.400, x.500, and MS Exchange based mail systems.

The Business and Information Network will allow access to the following:

SDC data retrievals and consolidated SDS
Data Displays (Note: No access to the RTPS command network)
Shuttle, GSE and Facility Drawings
OMI's
System Engineering reference material
IWCS (PRACA, WAVE, ARMS, SFC, CASPR)
SCAN/AERO
SECAS/Ground Operations LAN
Office LANS/WANS
World Wide Web/Internet
JSC and MSFC Data Systems
Xterminal Session to approved NASA Computer Systems (EDAMS, LSDN, etc.)
FTP permissions (will be guarded closely)
Other systems that become necessary (e.g. CCS PLC Controls/Monitor)

The Consolidated Data Stream will be made up of HUMS, PMS, Shuttle Data Stream, RPS, Data fusion, and any other data streams utilized in the launch complex area. These data streams will be merged together into a new Shuttle Data Stream. This data will be available on the network for monitoring purposes.

5.6.1. World Wide Web/Internet Access

The World Wide Web/Internet will be used for referencing historical information to assist in testing and troubleshooting. Most times this information will be local on a KSC intranet. However, information may be located at another NASA center or even a non-NASA location.

The system will provide the capability to restrict world wide web access to and from CLCS test sets. NASA security policy regarding World Wide Web/Internet Access will be followed.

5.7. Advisory Systems

There are several advisory systems deployed in CCMS Control Rooms. There are two general classifications: Real Time Data Processing applications and business enhancement applications.

The real-time applications are generally implemented on a high performance dedicated CPU/Platform (e.g. SUN workstation), have a system service that supports independent acquisition and buffering of the PC GOAL packetized data stream (to the FD Element) and other user unique system services (e.g. CRT Plotting and presentation display) and are user (or third party) developed and maintained.

In some cases, application specific tools are used for development (e.g. G2, CLIPS. etc.). Examples of these types of systems are:

Propulsion System Advisor (PSA)
Integrated APU Neural Network AI System (IAPU)
DPS-LCC Expert System (DLES)
Hydrogen Umbilical Monitoring System (HUMS)

The business enhancement applications are generally a suite of COTS tools with user unique information/data (e.g., Databases, system engineering reference material, Internet tools, etc.). An example of these applications are the RI-CD-ROM Wirelist Notebook computers and the SSME Avionics portable computers authorized by the Control Room Enhancement Review Team (CERT). Each advisory system will be reviewed for transition (if applicable) to the CLCS architecture.

As each existing advisory system is analyzed, a decision will be made to either integrate/port the existing code to the CLCS environment or to allow a CLCS ‘connect’ to the existing system. Alternately, a user applications server may be a needed resource in, for example, the SDC (for user unique business CPU targeted applications).

Since these systems are ‘advisory’ in nature (augment, do not replace certified systems for decision making) the user requires more flexibility with less demanding configuration control requirements. With CLCS, a construct is desirable whereby the user can provide his/her own application independent upon the rigor applied to the CLCS Application Software. Additionally this would also require that the CLCS provide a standardized packetized data acquisition and buffering API for the Business Systems platform.

5.8. Operational Intercommunication System

OIS control will not be integrated into CLCS. Existing OIS resources used in CCMS will be transferred to CLCS consoles. There must be space in the console layout to accommodate this OIS equipment.

5.9. Operational Television Displays

OTV will be integrated with CLCS. For viewing, a camera sequencing function will be provided. The user will have the ability to playback OTV, including a slow motion option. Each console command position will be capable of displaying two camera views on at least a 20 inch color screen. Camera views are to be displayed in "windows" which will be sizable and moveable. A shared control device will be used to switch views and change window attributes.

5.9.1. View Sequencing

Console operators will be able to view one sequencer OTV drop. This corresponds to two OTV views per console position. (Reference Figure 2-3 for OTV counts) The present capability to sequence through a maximum of 60 views will be maintained. The present ability to adjust the display time between 1 to 60 seconds will be maintained. The capability to define, store and recall up to twelve pre-programmed sequences will be available.

5.9.2. Camera Control

During testing, console operators will have direct control of camera functions (pan, tilt, zoom, focus and lights) through a graphical interface (except positions in the support modules and OMR/OSR areas). Indirect control of cameras via OTV/CCTV controller personnel (current method) will still be available when requested per OMI (i.e. for major tests, countdown). CLCS applications will be able to status and control camera functions. Camera control permissions will be based on system and test requirements as defined by the users and JYVO. Users will be able to define, store and recall Preset Camera Positions (e.g. press a button and have the camera point in specific direction). Preprogrammed camera movement will be available (e.g. press a button and have the camera automatically scan an area). Preprogrammed camera movement could be tied to Timing and Countdown (e.g. at a specific time in the count have a camera point in a specific direction). Positional feedback (e.g. azimuth and elevation) from cameras will be available but not displayed continuously.

5.9.3. Timing

A form of timing is required to maintain the capability to synchronize OTV, OIS, and CLCS data. Universal time and countdown overlays are not required since this data will be available on all CLCS displays.

5.9.4. Special Functions for Recorded Views

Consoles will have capability for real-time hard copy (to the Shared I/O color printer). Consoles will have capability to start and stop recording of camera views for short sequences (to the Shared I/O area). Video playback with the following options will be available in the Shared I/O Area: Pause, Slow Motion Playback, Fast Forward, Rewind, Frame by Frame Playback, Frame Capture w/ Color, and Hard Copy. JYVO will retain responsibility for archival and long duration recording. Long duration recording requests will be called out by OMI.

6. Operations Scenarios

This Operations Scenario section outlines how CLCS will be used in the two categories of Shuttle processing: 1) Integrated vehicle testing including launch, and 2) Flight element and GSE testing. It walks the reader through how primary users of CLCS will interact with the fully deployed and implemented system. Additional system level requirements may be derived from this section.

Assume for the sake of these scenarios that we begin at the point when the Master function has enabled command authority to a predetermined console command configuration for each of the control rooms. Software has already been developed in a configuration controlled software production facility, and has been promoted to a configuration controlled server. The OCRs have been configured by the master function; including loading and initializing the test set equipment with software. Test set components (console workstations, gateways, common equipment drives and network components) have been loaded from the configuration managed system.

6.1. Integrated Vehicle/Launch Environment

The most comprehensive activity at KSC is launch, therefore, more CLCS resources are used during this test than at any other time. Maximum redundancy in common equipment and network components are needed for safe operations. The prime control room and common equipment configuration associated with the launch test set are under configuration management from the time the vehicle becomes integrated through launch. Common equipment supporting the launch test set is dedicated solely to that test. Common equipment must be fail operational and console workstations fail safe. The workstations are supplied with alternating power feeds to allow an engineer to move to an adjacent workstation in the event of a failure on the prime workstation.

A system engineer arrives on station for integrated testing or launch carrying nothing but a headset. He notifies the test conductor on the OIS system that he is ready to perform his portion of the test. The engineer views a drop down menu on his command screen and selects from a list of system applications which ones will be needed to perform the system functional checkout. On the BIN the engineer accesses the test procedure that will be run. He also brings up the World Wide Web to view system schematics and reference material if the need for troubleshooting should arise. The engineer also configures OTV through a GUI interface on the BIN. He can pan, tilt, and zoom the cameras in the field from this graphical interface. The OTV images appear on a 20 inch high resolution digital flat screen monitor.

If the system engineer happens to be working a hazardous portion of the test, he will require, as a backup, hardwire safing capability to the associated GSE. The panel is versatile and can be configured for a variety of system safing applications. The hardwire panels are programmed and validated when the vehicle became integrated in the VAB and will not be changed until launch. The hardwire system provides the engineer with constant readout of its path integrity, wholly independent from CLCS.

The engineer is now ready to command his system. The application display shows a graphical depiction of the system or process. For example, he selects the desired component or value and with a click of a mouse invokes edit limits, commands, or viewer options to include; constraints, fusion parameters, and real time plot capabilities. System parameters can be shown on the screen in engineering units.

Because this is launch day, several engineers observe and consult on the test from a support module located behind the system engineer. They have access to the business information network and OTV and can assist the system engineer with data analysis and retrievals, or watch for system leaks on various OTV views. Other systems may opt for command network access instead of OTV in the support module. This access to the command network would allow engineers to monitor and control subsystems without the need for a larger console workstation. The engineers can communicate via OIS for formal call outs, but have the ability to conference if the need arises, due to their close proximity.

The control room layout is designed for multi-system integrated testing, which means the console workstations are organize in logical groupings (e.g. data systems, fluid systems, electrical systems, etc.) to support collaboration and conferencing by the team.

Test Directors and test conductors support integrated testing and launch from rows of specialized consoles differing from system consoles. These positions require less data viewing space and more space for legacy systems like expanded OIS, additional phones, Astronaut Communications panels, Area Warning System panels, etc. These test conductor workstations do have access to the BIN, RTCN, and OTV.

For integrated testing the prime room (either OCR 1 or OCR 2) is configured as stated above for testing. In the launch environment, in addition to the prime control room, the other OCR (either OCR1 or OCR 2) is configured to support engineering management's and system specialists' participation in the test.

Engineers and system specialists located in the off line control room for launch will be participating in the test from similar console workstations and support modules to those in the prime control room, however these workstations will not have commanding enabled. These engineers will have access to all the data and displays being used in the prime control room but for monitoring only. With the exception of command authentication and active hardwire safing panels, all other capability in the off line room remains the same as the prime room.

The mission management team, center management, and VIPs will be participating in launch activities from the OMR and OSR respectively. These rooms areas are in the prime control room as they are today. Each area is configured with support modules which house BIN workstations, phones, and OIS. Five large OTV monitors are positioned around each area to allow for unobstructed viewing.

Off center data users such as JSC MER and the HOSC at MSFC will have access to the Consolidated Data Stream. They will access the functional equivalent of PC-GOAL on the BIN implemented through an intranet client based interface.

6.2. Non Integrated/Multi-flow Environment

Non-integrated or multi-flow testing is more associated with the day-to-day activities at KSC and therefore requires the most efficient use of CLCS resources. Any of the three control rooms can be configured with multiple TCID loads. The intention is to process two or more non-integrated vehicles and their associated GSE out of a single control room. From landing through OPF processing an orbiter testing is accomplished in a flow zone environment. The room layouts are symmetrical in nature to clearly delineate the flow zones. Visual clues such as signs and color coding will also aid the operators in orienting themselves with flow zones. Because a control room supports multiple facilities, flight elements, and GSE, OIS must be configured to support multiple mission amps in the same room. Several predefined OIS configurations will be required to support this increased flexibility. As a vehicle nears completion of its non-integrated flow, it and its associated GSE would typically be relocated from a flow zone environment to a prime room in preparation for launch. At this point, only the single TCID for the integrated vehicle and its GSE would be permitted in the room until launch.

In addition to the room being divisible for multi-flows, the consoles are also flexible and configurable. Any console workstation can support any system testing. The number of workstations assigned to a flow zone is also variable. A flow zone will be expanded as testing needs increase and reduced when testing needs decrease. The test conductor, the master function, or any system granted command authentication authority can manage configuration of the flow zones and enable commanding at individual console workstations as the need arises. The system engineer requests a command position and the test conductor assigns a location and modifies the master console command authentication configuration as required to support that given day's testing requirements. The command authentication task requires that the test conductor have access to the command authority management software. This access is provided in two support modules located on the test conductor rows.

In addition to command authentication of control room workstations, the test conductor will also manage command and control authentication of local operations. Local command network drops would be provided at selected Shuttle processing sites (i.e., OPFs, VAB, MLPs, and Pads) to allow engineers to bring their CLCS testing to the vehicle. Engineers will have the capability to work on site with technicians for non-hazardous tests on a portable workstation. This type of testing could occur in the orbiter or at remote GSE locations such as the hydraulic pump units in the MLPs. It would apply to both integrated and non-integrated vehicle and flight element testing. It would enable the engineer to see and hear the hardware react when commands are sent as well as reduce the need for engineers supporting concurrently at the control room and the test site. In addition, future capability exists to add portable COTS test equipment to the local control workstations allowing data acquisition and storage through the SDC.

In a flow zone for powered vehicle testing CLCS will enable orbiter power up and monitoring from a single workstation. A single engineer will be capable of controlling, safing, and powering up/down an orbiter. This relieves the requirement for five engineers on station for the duration of power on testing. It is intended that advancements in application software and displays will allow significant consolidation in day- to-day multi-flow processing.

6.3. Simulation Environment

CLCS allows changes in the way simulations data is broadcast through the system. Links and gateways are no longer dedicated to simulation testing. This allows greater flexibility in scheduling and resource allocation for flight element processing during simulations. With the added flexibility offered by CLCS, integrated simulation can take place in a combination of an OCR, sections of another OCR with reduced management support, and office areas. Smaller multi-system simulations can be run from an IDE. Single system simulation can be performed from the desk top in an office environment.

The simulation system will support S0044 (Launch Countdown Simulation), S0056 (Cryogenic Loading Simulation), and S0066 (Hypergolic Loading Simulation) training. The simulation system will also support individual system training by providing the capability to load up either a single system or integrated model in a standalone training scenario.

7. Acronym List

A	
API	Application Program Interface
ARMS	Acquisition Resource Management System
ATE	Automated Test Equipment
ATM	Asynchronous Transfer Mode
B	
BIN	Business Information Network
C	
CCC	Complex Control Center
CCM	Command, Control and Monitor
CCP	Command, Control Processor
CCS	Complex Control System
CERT	Control Room Enhancement Review Team
Ch	Channel
CITE	Cargo Integration Test Environment
CLCS	Checkout and Launch Control System
COTS	Commercial Off-the-Shelf
CRT	Cathode Ray Tube
CASPR	Computer Aided Scheduling and Planning Resource
CSCI	Computer Software Configuration Item
D	
DBSAFE	Data Bank Shuttle Automated Function Executive
DCN	Display and Control Network
DDP	Data Distribution Processor
DFRC	Dryden Flight Research Center
DLES	DPS-LCC Expert System
DM	Display Manager
DPS	Data Processing System
E	
ECL	Environmental Control and Life Support System
ECLSS	Environmental Control and Life Support System
ECS	Environmental Control System
EDAMS	Engineering Data Access Management System
EIM	End Item managers
EPDS	Electrical Power Distribution System
ESA	Engineering Support Area
ESR	Engineering Support Request
ET	External Tank

F

FD	Function Designator
FR	Firing Room
FTP	File Transfer Protocol

G

GLS	Ground Launch Sequencer
GLS	Ground Launch Sequencer
GSE	Ground Support Equipment
GUI	Graphical User Interface
G/W	Gateway

H

HCI	Human Computer Interface
HIM	Hardware Interface Module
HMF	Hypergolic Maintenance Faciltiy
HOSC	Huntsville Operations Support Center
HUMS	Hydrogen Umbilical Monitoring System

I

I/O	Input/Output
IAPU	Integrated APU Neural network AI System
ID	Identification
IDE	Integrated Development Environment
ILMF	Intermediate Level Maintenance Facility
INTG	Integration
ISL	Instrumentation
IWCS	Integrated Work Control System

J

JSC	Johnson Space Center
JYVO	OTV Operator

K

KSC	Kennedy Space Center

L

LAN	Local Area Network
LC	Launch Control
LCC	Launch Control Center
LDB	Launch Data Bus
LOX	Liquid Oxygen
LPS	Launch Processing System
LRU	Line Replaceable Unit
LSDN	LPS Software Development Network

M

MDM	Multiplexer/Demultiplexer
MER	Mission Evaluation Room
MLP	Mobile Launcher Platform
MPCC	Maintenance Planning Control Center
MPTS	Minimum Peripheral Test Set
MSFC	Marshall Space Flight Center

N

NASA	National Aeronautics and Space Administration

O

OCR	Operations Control Rooms
OIS	Operational Intercom System
OPF	Orbiter Processing Facility
ORT	Operational Readiness Test
OS	Operating System
OSB	Operations Support Building
OSR	Operations Support Room
OTV	Operational Television

P

PCM	Pulse Coded Modulation
PFP	Programmable Function Panel
PFP	Programmable Function Panels
PLC	Primary Logic Control
PMS	Permanent Measurements System
PRACA	Problem Reporting And Corrective Action
PRCBD	Program Requirements Control Board
PSA	Propulsion System Adviser

R

RCL	Reactive Control Logic
RCN	Requirements Change Notice
RPS	Record and Playback System
RTCN	Real Time Critical Network
RTPS	Real Time Processing System

S

SAIL	Shuttle Avionics Integration Laboratory
SCAN	Shuttle Connector Activity Network
SCID	System Configuration Identifier
SDC	Shuttle Data Center
SDE	Software Development Environment
SECAS	Shuttle Engineering Computer Application System
SFC	Shop Floor Control
SFOC	Space Flight Operations Contract
SSCR	Simulation Software Change Request
SSME	Space Shuttle Main Engine
SSPF	Space Station Processing Facility

T	
TBD	To Be Decided
TC	Test Conductor
TD	Test Director
TCID	Test Configuration Identifiers
V	
VAB	Vehicle Assembly Building
W	
WAD	Work Authorizing Document
WAN	Wide Area Network
WAVE	WAD Authoring and Validation Environment